

Paperless ngx

- Installation
 - Installation in Docker
 - Ersteinrichtung

Installation

Installation in Docker

Beschreibung:

Hier wird ein Docker container erzeugt, das Volumes benötigt. Weil die ordner spezielle Benutzerrechte haben.

Installation

Vorraussetzung

Dockerinstallation

```
apt install docker.io docker-compose
```

Die Environment Datei

Hier werden Sprache Zeitzone und das Datenbank Passwort festgelegt.
nicht vergessen die URL auch festzulegen entweder Domain oder ip aber mit https:// davor

docker-compose.env

```
# The UID and GID of the user used to run paperless in the container.  
#USERMAP_UID=1000  
#USERMAP_GID=1000  
  
# Additional languages to install for text recognition.  
#PAPERLESS_OCR_LANGUAGES=tur ces  
  
# Paperless-specific settings  
PAPERLESS_URL=https://paperless.example.com oder https://ip
```

```
#PAPERLESS_SECRET_KEY=change-me
#PAPERLESS_TIME_ZONE=America/Los_Angeles
#PAPERLESS_OCR_LANGUAGE=eng
#PAPERLESS_FORCE_SCRIPT_NAME=/PATHPREFIX
#PAPERLESS_STATIC_URL=/PATHPREFIX/static/

# Redis settings for Paperless
PAPERLESS_REDIS=redis://broker:6379

# Database configuration for MariaDB
MARIADB_HOST=paperless
MARIADB_DATABASE=paperless
MARIADB_USER=paperless
MARIADB_PASSWORD=paperless
MARIADB_ROOT_PASSWORD=paperless

# Paperless webservice environment settings
PAPERLESS_DBENGINE=mariadb
PAPERLESS_DBHOST=db
PAPERLESS_DBUSER=paperless
PAPERLESS_DBPASS=paperless
PAPERLESS_DBPORT=3306
```

Deutsch angepasste env Datei

```
# The UID and GID of the user used to run paperless in the container.
#USERMAP_UID=1000
#USERMAP_GID=1000

# Additional languages to install for text recognition.
PAPERLESS_OCR_LANGUAGES=deu

# Paperless-specific settings
PAPERLESS_URL=https://paperless.example.com oder https://ip
#PAPERLESS_SECRET_KEY=change-me
PAPERLESS_TIME_ZONE=Europe/Berlin
PAPERLESS_OCR_LANGUAGE=deu
#PAPERLESS_FORCE_SCRIPT_NAME=/PATHPREFIX
#PAPERLESS_STATIC_URL=/PATHPREFIX/static/
```

```
# Redis settings for Paperless
PAPERLESS_REDIS=redis://broker:6379

# Database configuration for MariaDB
MARIADB_HOST=paperless
MARIADB_DATABASE=paperless
MARIADB_USER=paperless
MARIADB_PASSWORD=paperless
MARIADB_ROOT_PASSWORD=paperless

# Paperless webserver environment settings
PAPERLESS_DBENGINE=mariadb
PAPERLESS_DBHOST=db
PAPERLESS_DBUSER=paperless
PAPERLESS_DBPASS=paperless
PAPERLESS_DBPORT=3306
```

Die docker-compose Datei

```
version: "3.4"
services:
  broker:
    image: docker.io/library/redis:7
    restart: unless-stopped
    volumes:
      - redisdata:/data
    env_file: docker-compose.env

  db:
    image: docker.io/library/mariadb:10
    restart: unless-stopped
    volumes:
      - dbdata:/var/lib/mysql
    env_file: docker-compose.env

  webserver:
    image: ghcr.io/paperless-ngx/paperless-ngx:latest
    restart: unless-stopped
    depends_on:
```

```
- db
- broker
ports:
- "8000:8000"
volumes:
- data:/usr/src/paperless/data
- media:/usr/src/paperless/media
- ./export:/usr/src/paperless/export
- ./consume:/usr/src/paperless/consume
env_file: docker-compose.env
```

```
volumes:
data:
media:
dbdata:
redisdata:
```

Nun das Shell script für die SSL cert und die nginx conf

```
nano create_ssl_cert.sh
```

Inhalt:

```
#!/bin/bash

# Directory for SSL certificates
mkdir -p ./cert

# Generate SSL certificates
openssl req -x509 -nodes -days 36500 -newkey rsa:2048 -keyout ./cert/server.key -out ./cert/server.crt -subj
"/C=US/ST=Denial/L=Springfield/O=Dis/CN=yourdomain.com"

echo "SSL certificates generated and stored in ./cert directory."
```

Nun das shell script ausführbar machen

```
chmod +x create_ssl_cert.sh
```

Dann ausführen

```
./create_ssl_cert.sh
```

Nun die nginx.conf erstellen

```
nano nginx.conf
```

Inhalt

```
worker_processes 1;

events {
    worker_connections 1024;
}

http {

    # Adjust as required. This is the maximum size for file uploads.
    # The default value 1M might be a little too small.
    client_max_body_size 1000M;

    server {

        listen 443 ssl;
        #server_name yourdomain.com;

        ssl_certificate /etc/nginx/cert/server.crt;
        ssl_certificate_key /etc/nginx/cert/server.key;

        location / {

            # Adjust host and port as required.
            proxy_pass http://webserver:8000/;

            # These configuration options are required for WebSockets to work.
            proxy_http_version 1.1;
            proxy_set_header Upgrade $http_upgrade;
            proxy_set_header Connection "upgrade";

            proxy_redirect off;
            proxy_set_header Host $host;
```

```
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header X-Forwarded-Host $server_name;
add_header Referrer-Policy "strict-origin-when-cross-origin";
}
}
}
```

nun mit

```
docker-compose up -d
```

den kram starten

Danach kann über Port <https://ip>

Paperless aufgerufen werden



Bitte melden Sie sich an.

Anmelden

Benutzer erstellen

Nun haben wir die Anmeldeseite aber noch keinen Benutzer.

Dazu rufen wir:

```
docker-compose run --rm webserver createsuperuser
```

Nun können wir einen Benutzernamen festlegen

Hier lassen wir paperless

Dann eine Email angeben

Dann das Passwort angeben und nochmals bestätigen

```
Creating paperless_webserver_run ... done
```

```
Paperless-ngx docker container starting...
```

```
Installing languages...
```

```
Get:1 http://deb.debian.org/debian bookworm InRelease [151 kB]
Get:2 http://deb.debian.org/debian bookworm-updates InRelease [55.4 kB]
Get:3 http://deb.debian.org/debian-security bookworm-security InRelease [48.0 kB]
Get:4 http://deb.debian.org/debian bookworm/main amd64 Packages [8786 kB]
Get:5 http://deb.debian.org/debian bookworm-updates/main amd64 Packages [13.8 kB]
Get:6 http://deb.debian.org/debian-security bookworm-security/main amd64 Packages [155 kB]
Fetched 9209 kB in 2s (5377 kB/s)
Reading package lists... Done
Package tesseract-ocr-deu already installed!
Creating directory scratch directory /tmp/paperless
Adjusting permissions of paperless files. This may take a while.
Waiting for MariaDB to start...
Waiting for Redis...
Connected to Redis broker.
Apply database migrations...
Operations to perform:
  Apply all migrations: account, admin, auditlog, auth, authtoken, contenttypes, django_celery_results,
documents, guardian, paperless, paperless_mail, sessions, socialaccount
Running migrations:
  No migrations to apply.
Running Django checks
System check identified no issues (0 silenced).
Executing management command createsuperuser
Username (leave blank to use 'paperless'):
Email address: test@test.de
Password:
Password (again):
Superuser created successfully.
```

Nun können wir uns einloggen:

Die Startseite :

- Startseite
- Dokumente

VERWALTUNG

- Korrespondenten
- Tags
- Dokumenttypen
- Speicherpfade
- Benutzerdefinierte Felder
- Arbeitsabläufe
- E-Mail

ADMINISTRATION

- Einstellungen
 - Konfiguration
 - Benutzer & Gruppen
 - Dateiaufgaben
 - Protokolle
 - Dokumentation
- Paperless-ngx 2.7.2

Startseite

Hallo paperless, willkommen bei Paperless-ngx

Paperless-ngx wurde erfolgreich gestartet!

Sie sind bereit, Dokumente hochzuladen! Entdecken Sie die verschiedenen Funktionen dieser Webanwendung auf eigene Faust oder starten Sie eine kurze Tour mit der folgenden Schaltfläche.

Mehr Details zur Verwendung und Konfiguration von Paperless-ngx sind immer in der [Dokumentation](#) verfügbar.

Vielen Dank, dass Sie Teil der Paperless-ngx-Community sind!

[Tour beginnen →](#)

Statistiken

Dokumente insgesamt: 0

Zeichen insgesamt:

Neue Dokumente hochladen

Dokumente irgendwo hinziehen oder

[Dateien auswählen](#)

Installation

Ersteinrichtung

Beschreibung:

Da wir nun ein laufendes System mit einem benutzer haben.
nun die grundkonfiguration.
Dokumententypen und Speicherziele

Ersteinrichtung:

Korrespondenten einrichten.
Das Sind Personen oder Gesellschaften, z.b Allianz, generali Hausverwaltung etc.

Dokumententypen hinzufügen.

Z.b Haftpflichtversicherung, Kaskoversicherung, Nebenkostenabrechnung

Tags einrichten. Diese besagen eine Ober Kathegoruie z.b Auto, Wohnung, Lebensmittel, Strom, Gas, Wasser etc.